

Trabajo RNP - 2019 cuat. 2

Patrones en registros electromiográficos de insectos

G. Sebastián Pedersen (sebasped@gmail.com)

Héctor Salas (hecsalms@gmail.com)

Ma 03-Dic-2019

1. Presentación del problema

Se tienen registros electromiográficos de insectos que se desean clasificar de acuerdo a dos situaciones: el insecto está comiendo vs. el insecto no está comiendo. El setup de la medición se puede ver en la figura 1.

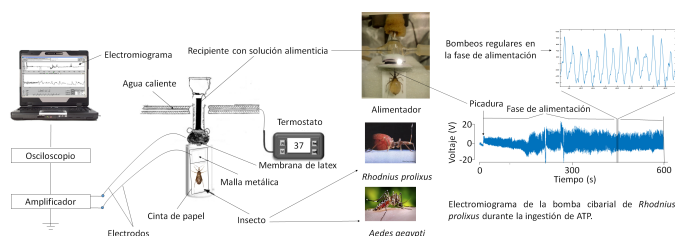


Figura 1: Setup de la medición.

Se trabajaron con datos de laboratorio reales. Una serie temporal típica se puede ver en las figuras 2 y 3.

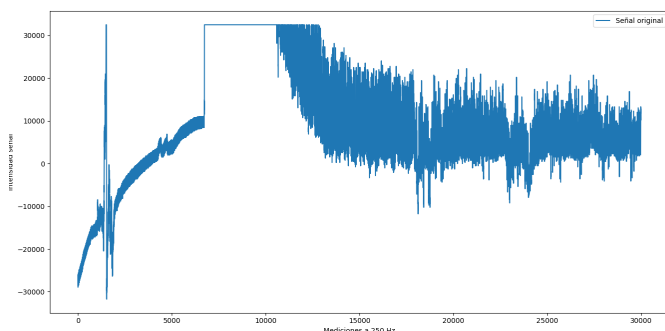


Figura 2: Serie de datos temporal típica.

Como la serie temporal consta de la deriva típica proveniente de mediciones con aparatos electrónicos, como preprocesamiento a la señal para quitar esta deriva se utilizó el algoritmo BEADS (ver [1]). La serie luego de remover la deriva se puede ver en las figuras 4 y 5.

Posteriormente se dividió la serie temporal en pequeñas ventanas, cada una clasificada por Sí/No respecto al insecto comiendo. Se probaron con ventanas de 6, 1 y 0.5 segs. La figura 6 ilustra la situación.

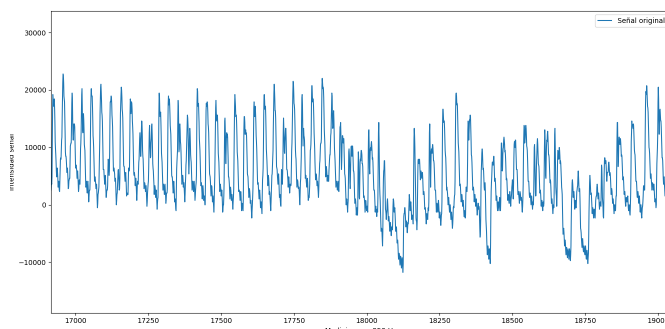


Figura 3: Serie de datos temporal típica.

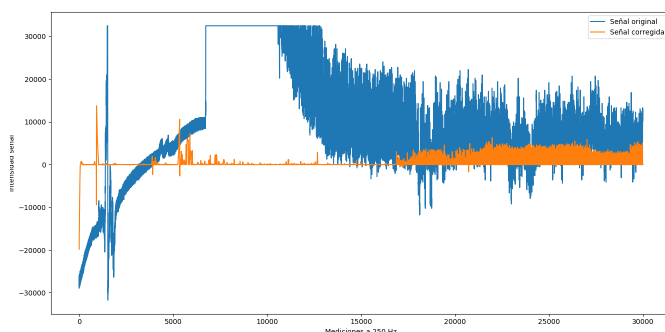


Figura 4: Serie de datos temporal típica (azul) y corregida por BEADS (naranja).

2. Redes y resultados

Se probaron varias redes convolucionales, para los distintos anchos de ventanas. Por suerte se contaban con muchos datos (3000 para ventanas de 6 segundos y 18 mil para de 1 segundo).

Se particionaron los datos en 80/20 % train/test respectivamente, y se mezclaron aleatoriamente. Además siempre se trabajó con datasets previamente balanceados 50/50 % para cada clase respectivamente.

En general los resultados, medidos en accuracy, obtenidos fueron aceptables, de 90 % en adelante. El mejor resultado se obtuvo en ventanas de tamaño 1 seg. con una ConvNet de 3 convoluciones, 2 max poolings, y 3 FCs, el cual fue de un accuracy del 95-96 %. Además se normalizaron previamente los datos de entrada.

La figura 7 ilustra el resultado respecto a épocas de

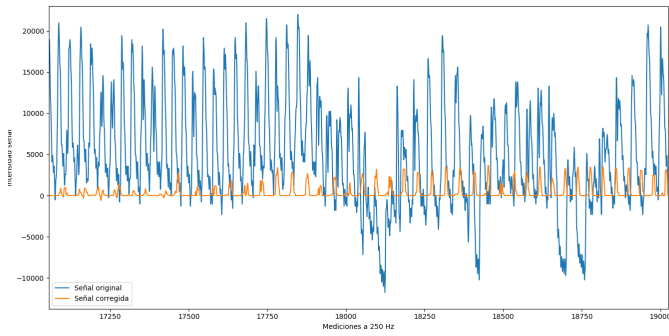


Figura 5: Serie de datos temporal típica (azul) y corregida por BEADS (naranja).

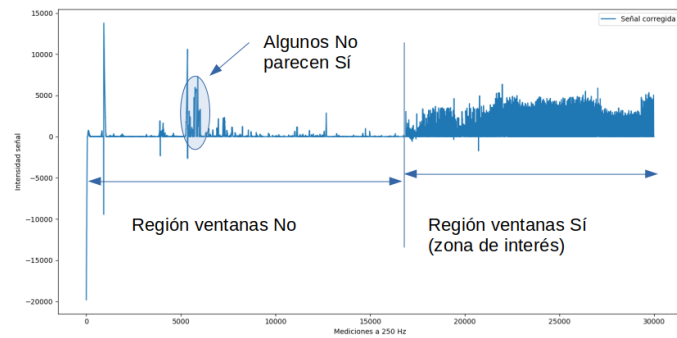


Figura 6: Regiones utilizadas para generar las ventanas de clasificación. Cada región consta de muchas ventanas.

entrenamiento.

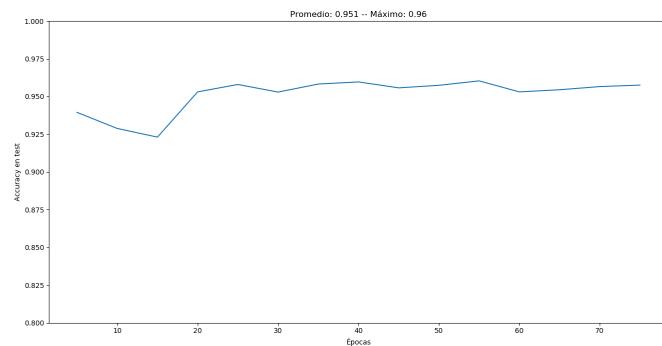


Figura 7: Mejor resultado de clasificación de ventanas, respecto al accuracy.

Finalmente, y como prueba rápida de último momento, se probó clasificar utilizando las FFTs de las ventanas (parte real y compleja). El resultado también fue satisfactorio, como muestra la figura 8, con un accuracy del 91-92 %.

Posteriormente se probó otra red convolucional pero dilated, obteniéndose un accuracy algo mayor al 96 %. Esta red se entrenó por 30 épocas, y en la figura 9 se muestra el accuracy respecto a cada dato de test (800 en total).

Y finalmente se probó una red recurrente, obteniéndose un accuracy del 93 % como muestra la figura 10.

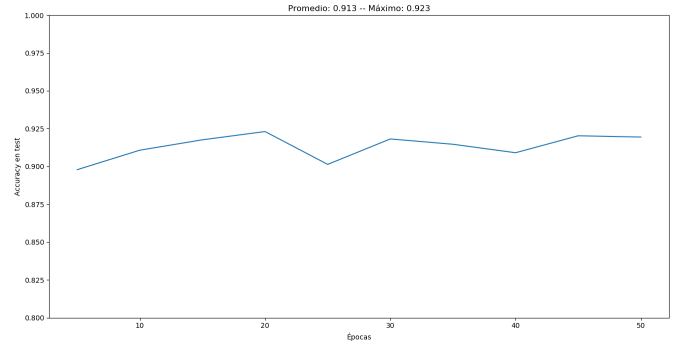


Figura 8: Mejor resultado de clasificación de ventanas por FFTs, respecto al accuracy.

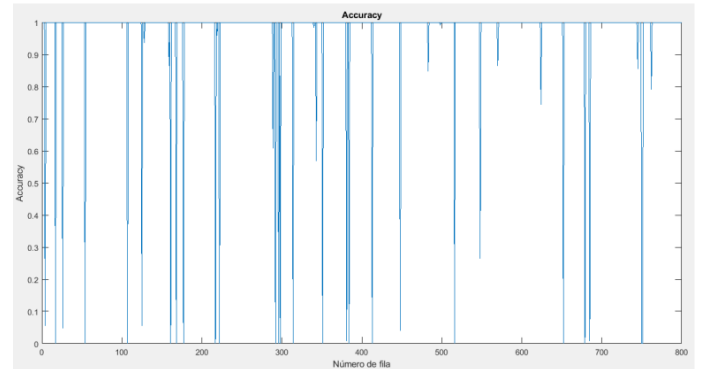


Figura 9: Resultado clasificación por accuracy con la convolucional dilated.

3. Cómo seguir

Quedaron varias cuestiones a probar o mejorar, a saber:

- Probar con otras arquitecturas/parámetros para las CNNs. En particular achicar el tamaño de los filtros en las convolucionales.
- Probar otras Recurrentes.
- Input con series temporal + FFTs.
- Ensamblar.
- Barrer el tamaño de la ventana más exhaustivamente.
- Barrer el tamaño de mini-batch más exhaustivamente.
- Reconocer más patrones en los registros, por ejemplo:
 - Cantidad y tiempo de cada evento de picado/no-picado.
 - Tiempo, frecuencia, amplitud media y cantidad de bombeos.
- Utilizar la matriz de confusión como métrica de performance, para detectar mejor dónde está fallando la clasificación.

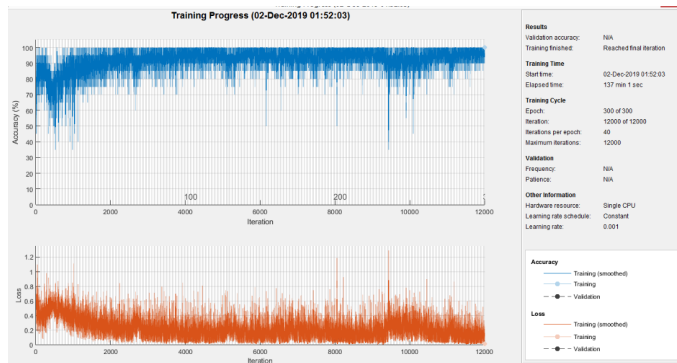


Figura 10: Resultado clasificación por accuracy con la recurrente.

4. Arquitecturas

4.1. ConvNets

Las convolucionales se utilizaron variantes de:

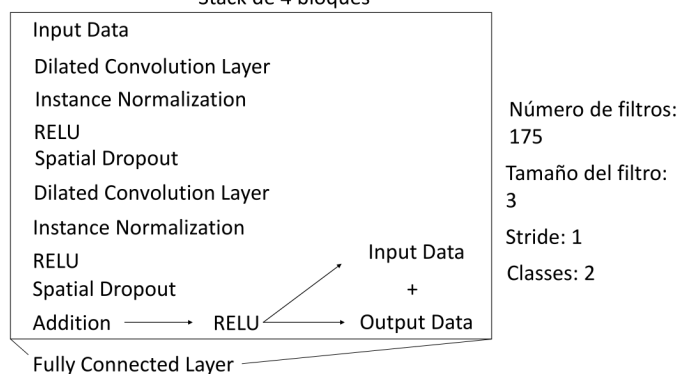
- Input: ventanas de cant. seg. corresp. (por 250 Hz de sampleo).
- Convolutional 1D de 5x60 (stride=1, padding=30) + ReLu
- Convolutional 1D de 10x30 (stride=2, padding=30)
- Max Pooling de 2 + ReLu
- Convolutional 1D de 20x15 (stride=2, padding=30)
- Max Pooling de 2 + ReLu
- FC de 580 a 300 + tanh
- FC de 300 a 100 + tanh
- FC de 100 a Sí/No

Las variantes esencialmente incluyeron pruebas sin la última convolucional, y también sin al FC intermedia.

4.2. ConvNet Dilated

Para la convolucional dilated la arquitectura fue:

Stack de 4 bloques



4.3. Recurrente

Y para la recurrente la arquitectura fue:

Input Data
LSTM Layer
Instance Normalization
Fully Connected
Softmax Layer
Classification Layer

Número de Unidades Ocultas:
100

Referencias

- [1] Xiaoran Ning; Ivan W.Selesnick; Laurent Duval;
Chromatogram baseline estimation and denoising using sparsity (BEADS);
<https://doi.org/10.1016/j.chemolab.2014.09.014>